

Genetic networks, non-monotonic logic and SAT Algorithms

Andrei Doncescu², Vincent Risch¹, Sylvain Sene¹, Pierre Siegel¹

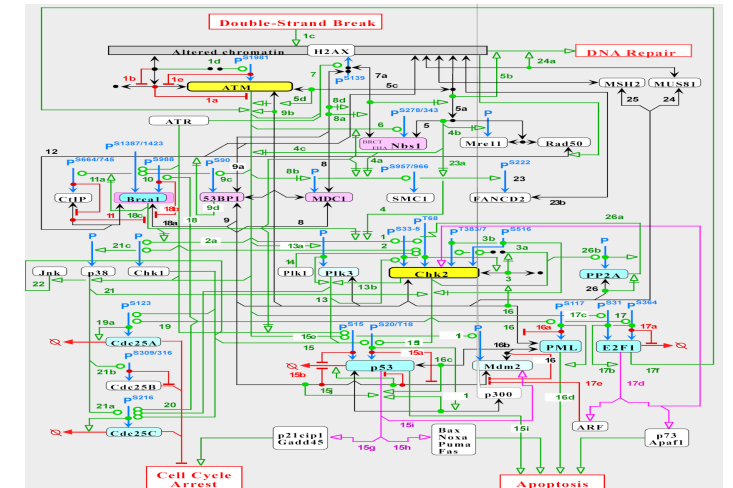
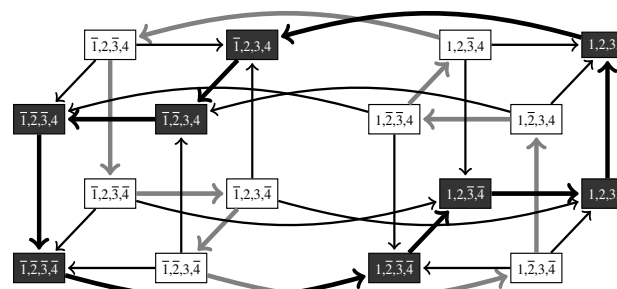
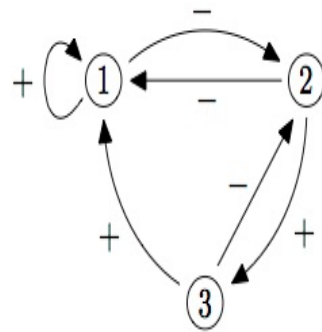
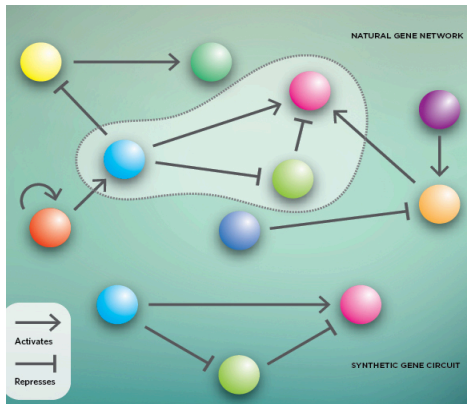
1 LIS, Aix-Marseille-Université 2 LAMIA, Université des Antilles, La Guadeloupe

- 1 Genetic networks.
- 2 Boolean Dynamical Systems (BDS).
- 3 Nonmonotonic logics, Hypothesis logic \mathcal{H} .
- 4 Representation of genetic networks in Hypothesis Logic
- 5 Links with BDS.
- 6 SAT based algorithms.

1 -Biological networks – Genetic networks - Boolean networks

A biological network is any network that applies to biological system : genetic regulatory networks, DNA–protein interaction networks, Metabolic networks, Signaling network ...

- A **genetic network** represents how the proteins/genes interact in a cell.
 - Techniques which are the most used in Network Modeling are **boolean networks**, bayesian belief network and metabolic network modeling methods...
 - From a logical point of view, a biological network is a set of interacting elements changing along discrete time.
- **Booleans networks** are well adapted for genetic networks:
 - *It is considered that genes are on-off switches which do not act independently.*
 - *The **expression** of one gene (**activation** or **inhibition**) modulates the expression of another ones genes.*



1 - The study of genetic networks is interesting regarding knowledge representation and AI

- Interactions appear as a form of ***causality***; as such we **expect to model by using logical inferences**.
- What we learn arises largely from experiments: **we know only a small part of the interactions**:
 - The knowledge can be ***revisable, uncertain, multi-sources, contradictory*** and even ***false***.
 - The use of classical logic is inadequate in this context because it **cannot deal with inconsistencies**.
- What we learn arises largely from **long and expensive experiments**.
 - For an initial experiment 'in-silico', we must complete the information: ***abduction***.
- Question of **algorithmic complexity** (\sum_2^P) provide algorithms with reasonable calculation times in practice.

Questions studied in AI since the 1970s, especially by the use of ***nonmonotonic logics*** and techniques derived from ***constraint programming***.

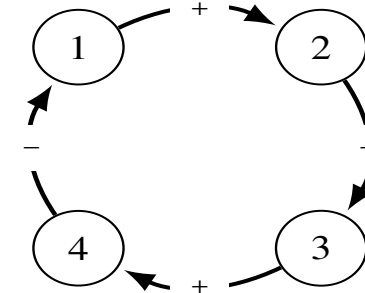
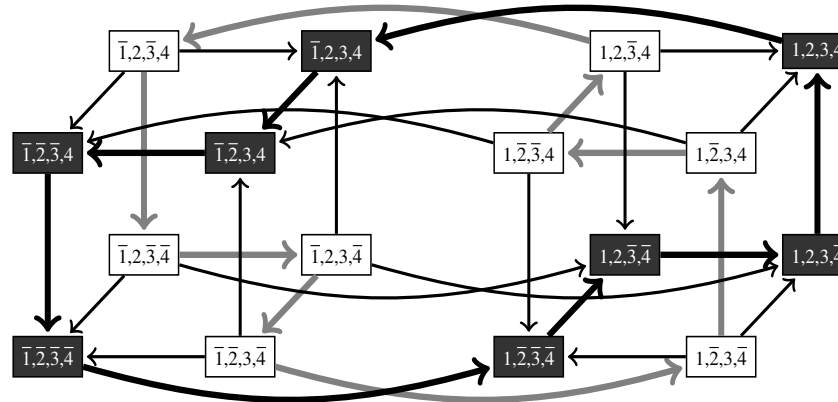
2 - Boolean Dynamical Systems (BDS)

- A **BDS** describes the evolution, over a discrete time, of interactions in a finite network E of n **entities**. This evolution is the **dynamic** of the BDS.
- A dynamic request to define an **update**:
 - **Asynchronous update**: at the time t only one entity is updated.
 - Fully synchronous (parallel) update: all the entities of the network are updated simultaneously.
 - ... and countless other mixed updates.
- A BDS can be represented by a **fonction f** , by a **transition graphs (TG)** or by an **interaction graph (IG)**

The exact links between these representations is still an open problem - Nonmononic logics could help

Negative circuit of size 4

$$f(x) = f(x_1, x_2, x_3, x_4) \\ = (\neg x_4, x_1, \neg x_2, x_3)$$



function

Asynchronous Transition graph (ATG)

Interaction Graph- IG 4

2 - Réseaux de gène et Systèmes Dynamiques Discrets (SDD)

- An **entity** i represents a protein/gene, x_i is the concentration of i .
- An interaction gives conditions for a set of genes to increase or decrease the concentration of a protein.
- If the concentration $x_i \in \{0,1\}$ we have a Boolean Dynamic System (BDS) .
 x_i ($x_i = 1$) denotes the presence of i , $\neg x_i$ ($x_i = 0$) denotes the absence of i
- i is associated with a propositional variable x_i

2 BDS - Asynchronous Transition Graph (ATG)

- Let $V = \{1, \dots, n\}$ a set of **entities** represented by n boolean variables.
- $X = \{0,1\}^n$ is a **configuration space**.
- $f : X \rightarrow X$ is the **global transition function** of a BDS, such that:
 $x = (x_1, \dots, x_n) \rightarrow f(x) = (f_1(x), \dots, f_n(x))$ where each function $f_i : X \rightarrow \{0,1\}$ is a **local transition function** that give the evolution of the entity i over time.

The **dynamic** of f is given by its **asynchronous transition graph** $ATG(f) = (X, T(f))$, a digraph whose vertex set is the **configuration space** and arc set is the **set of asynchronous transitions** such that:

$$T(f) = \{(x,y) \in X^2 \mid x \neq y, x = (x_1, \dots, x_i, \dots, x_n), y = (x_1, \dots, x_{i-1}, f_i(x), x_{i+1}, \dots, x_n)\}.$$

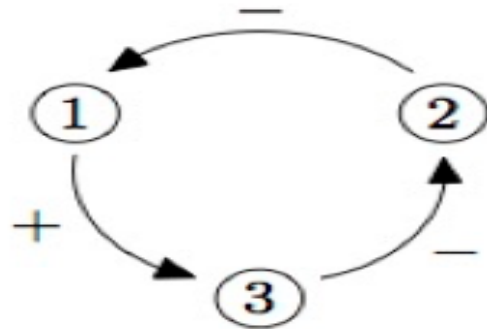
- Therefore, if $(x,y) \in T(f)$, x and y differ exactly by one element; the transition is **unitary**.

For logic a configuration is a propositional interpretation.

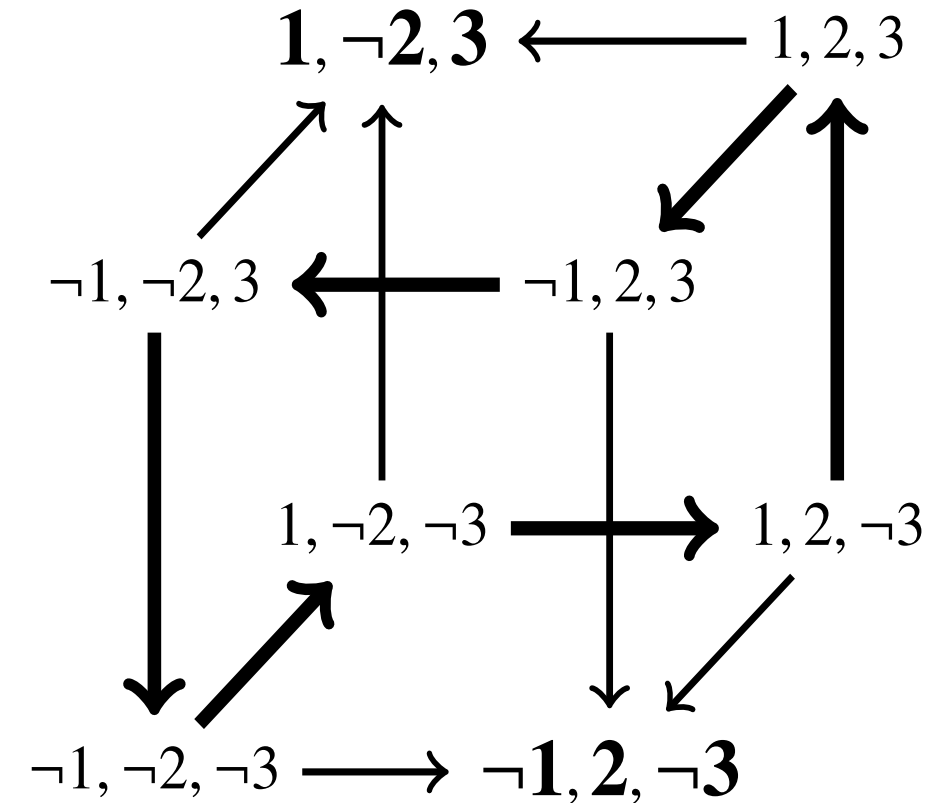
- Let $V = \{1, 2, 3\}$ the set of entities $X = \{0, 1\}^3$ the configuration space
- $F : X \rightarrow X$ is the global transition function.

$$f(x) = f(x_1, x_2, x_3) = (\neg x_2, \neg x_3, x_1)$$

- The **ATG** of f is a graph of $2^3 = 8$ vertices.
- Each vertex is a **configuration**, edges are the **transitions**.
- As the graph is **Asynchronous**, an arc reverses a single element of a configuration: *We do not "cross" the hypergraph*
- An **orbit** is a "walk" inside the graph.



Interaction graph



Asynchronous Transition Graph

2 BDS: Positive circuit of size 3

➤ 2 symmetrical **stable configurations**:

$x_1 = (1, \neg 2, 3)$ and $x_2 = (\neg 1, 2, \neg 3)$

Representation by Minimal models of Prolog(1972),

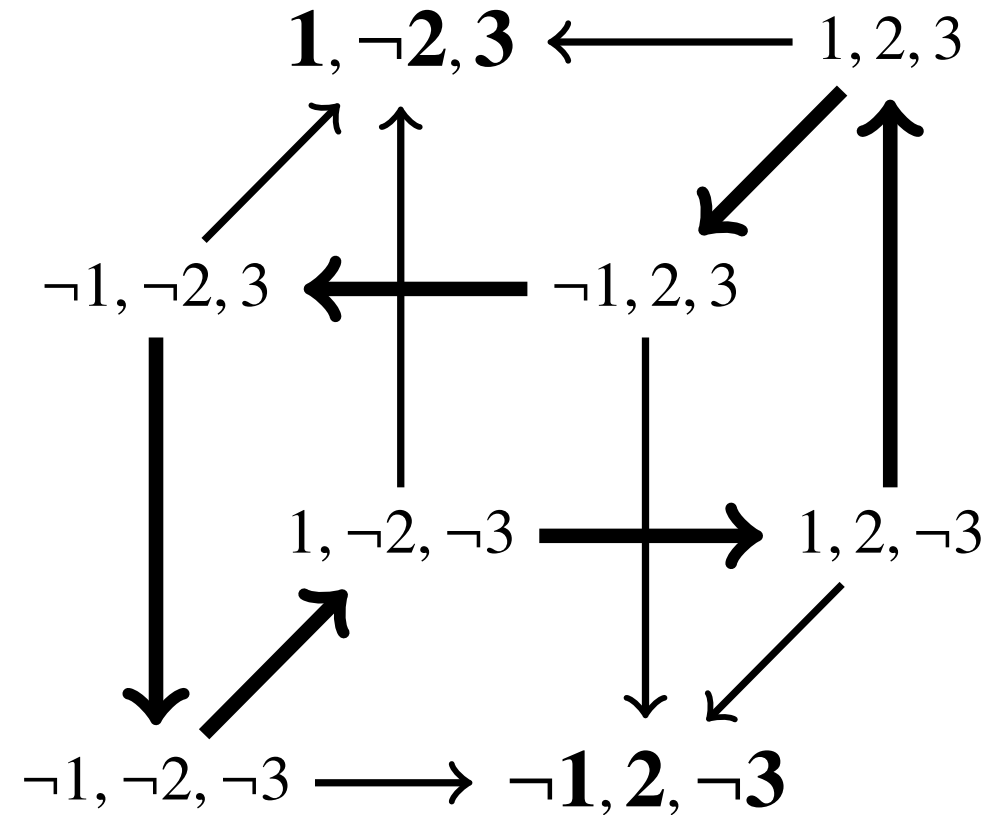
Extensions of default logic (1980),

Preferential Models (1981),

Stable models (1989) ...

➤ An **unstable cycle** in bold.

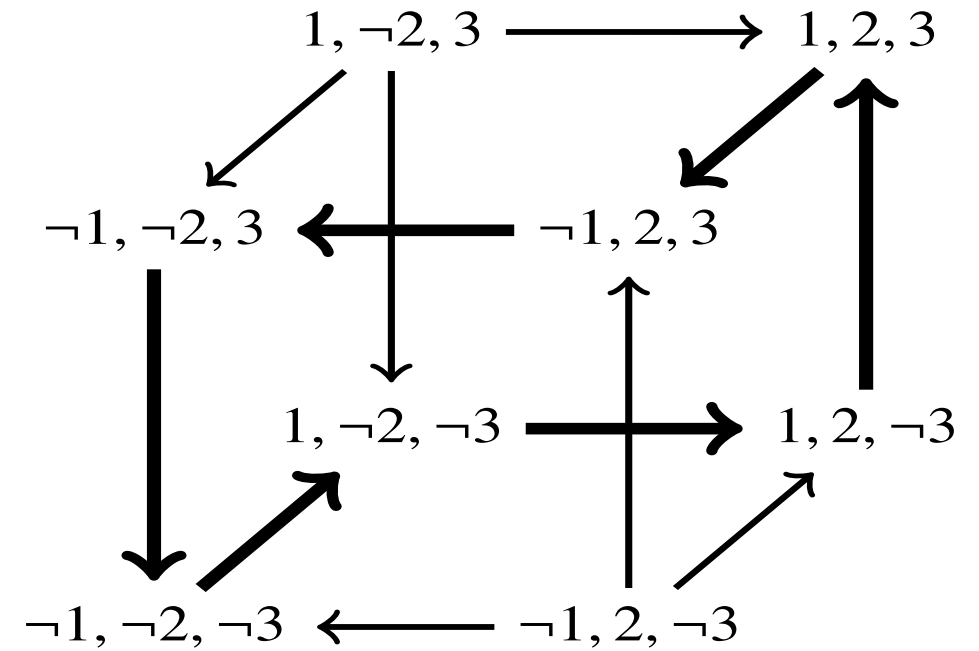
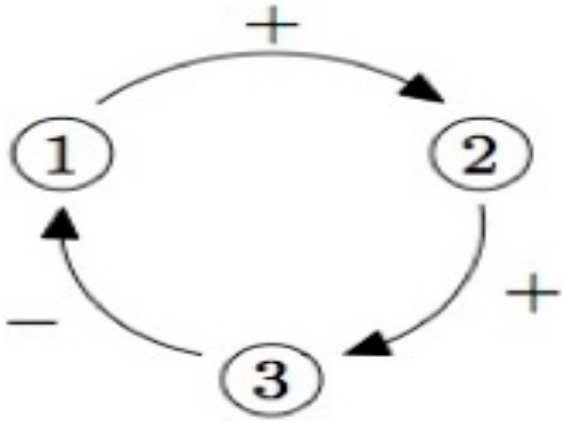
- *It is to leave a stable configuration*
- *The cycle is **unstable** because we can leave at any moment (here join a stable configuration).*
- *We obtain an infinity of orbits*



Question: Logic representation of cycles ?

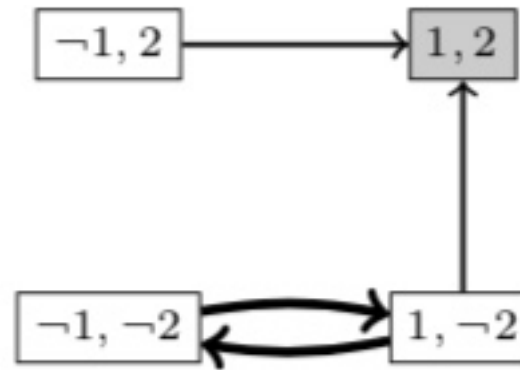
Function $g(x_1, x_2, x_3) = (\neg x_3, x_1, x_2)$:

- A **stable cycle** in bold
- All orbits join the stable cycle and never come out.



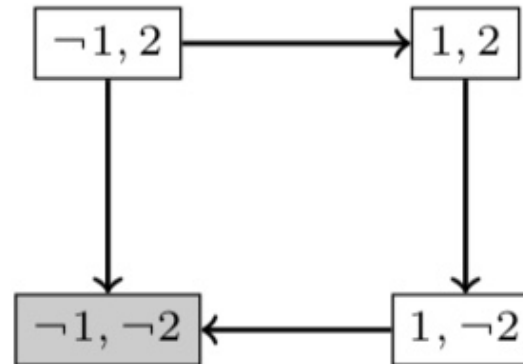
2-Examples

$$h(x_1, x_2) = (\neg x_1 \vee x_2, x_1 \vee x_2)$$



1	2	h_1	h_2	degree
0	0	1	0	1
0	1	1	1	1
1	0	0	1	2
1	1	1	1	0

$$k(x_1, x_2) = (x_2, \neg x_1 \wedge x_1 \wedge x_2)$$



1	2	k_1	k_2	degree
0	0	0	0	0
0	1	1	0	2
1	0	0	0	1
1	1	1	0	1

ATG

Truth table

2 - Return to cells: General central results

- **Thomas's conjectures:**

- The presence of several attractors makes it possible to model, using dynamic behaviors, the cellular specialization (1973).
- The interaction graph of a SDS must contain a positive circuit to have several stable configurations (necessary condition) - 1981
- The interaction graph of a SDS must contain a negative circuit to have a stable cycle (necessary condition) 1981

Example: The Arabidopsis thaliana plant, modeled in SDB, whose dynamic behaviors admit in particular four stable configurations that correspond to the gene expression patterns of floral tissues: sepals, petals, stamens and carpels.

- **Robert's theorem (1986):**

- If the interaction graph is acyclic (DAG), then f converges to a single stable configuration

- **Rémy, Mossé, Chaouiya, Thierffry (2008):**

- An asynchronous positive circuit of size n has 2 stable configurations
- An asynchronous negative circuit only allows a stable cycle of length $2n$

We will find these properties with the translation into hypothesis logic.

3 - Nonmonotonic logics.

- « Classical » Logics: Propositional Calculus , First Order logic , Modal logics ...) are **monotonics**:

If $A \vdash B$ then $A \cup A' \vdash B$

If A infers B , then $A \cup A'$ infers B again.

If knowledge increases/decreases then the deductions increase/decrease.

- Mathematics is based on monotony: theorems are « accumulated ».
- In real life (and also in science) information is very often incomplete, uncertain, revisable, contradictory, false, multi-source, alternative) ...
- Biological systems:
 - Knowledge incomplete, uncertain, revisable, contradictory, false, multi-source, alternative ...
 - It is necessary to complete the information (in-silico experiences): *abduction*.
 - Algorithmic complexity in \sum_2^P

3 - Non Monotonic logic: “history”

- **Plato: *Allegory of the cave*** (390 bbc): Group of people are chained in a cave, facing a blank wall. They watch shadows projected on the wall from objects passing in front of a fire behind.

....

- **Prolog and *negation as failure* : not** (Colmerauer, Kowalski, Roussel - 1972).

$A :- \text{not}(B), C.$ “ *A is proved if B is not proved and C is proved*

`not (x) :- x, !, fail.`

`not (x) .`

- Closed world assumption (R. Reiter 1978), **Circumscription**(J. McCarthy 1980), Preferential models (G. Bossu, PS 1981 ; Shoam 1983).
- **Default Logic** (R. Reiter 1980), *ASP* (Marek and Truszczyński, Gelfond, Lifschitz-89).

.....

3 - Default Logic (Reiter 1980)

- **Default theory** $\Delta = (D, W)$:

W is a set of closed formulas of first order logic (FOL).

D is a set of defaults, **specific inference rules**, of the form $d = \frac{A : B}{C}$

« If A is proved, if B is consistent with all that will be proved, then C is proved »

- A set E of formulas of FOL is an **extension** of Δ iff:

$$(1) E = \bigcup E_i \quad (2) E_0 = Th(W) \quad (3) E_{i+1} = Th(E_i \cup \{C \mid \frac{A : B}{C} \in D, A \in E_i, \neg B \notin E\})$$

« We use as many defaults as possible »

- An extension is a fixed point.

- Extensions can represent stable configurations, « minimal models » of Prolog*, stable model, ASP**

- **Problems:**

- Even if W is consistent, there are default theories that have no extension (because of $\neg B \notin E$).

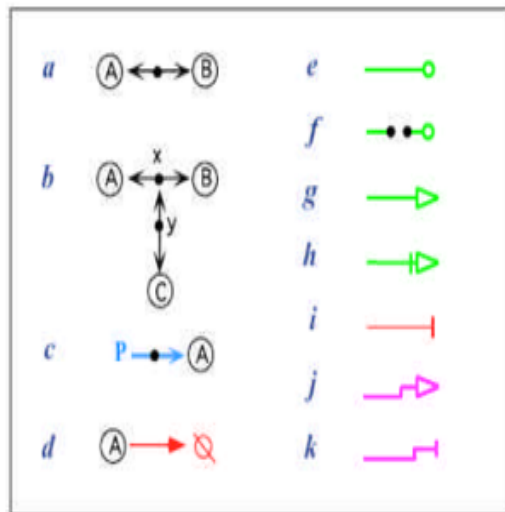
- DL is not cumulative, extensions calculation algorithms are not constructive.

*A Prolog clause (Colmerauer, Roussel 1972) $C :- A, \text{not}(B)$. is « translated » by $d = \frac{A : \neg B}{C}$

** An ASP rule (Marek and Truszczyński, Gelfond, Lifschitz-89..) $C :- A, \text{not } B$ is « translated » by $d = \frac{A : \neg B}{C}$

Cancer – P53 network model

How to stop cancer?

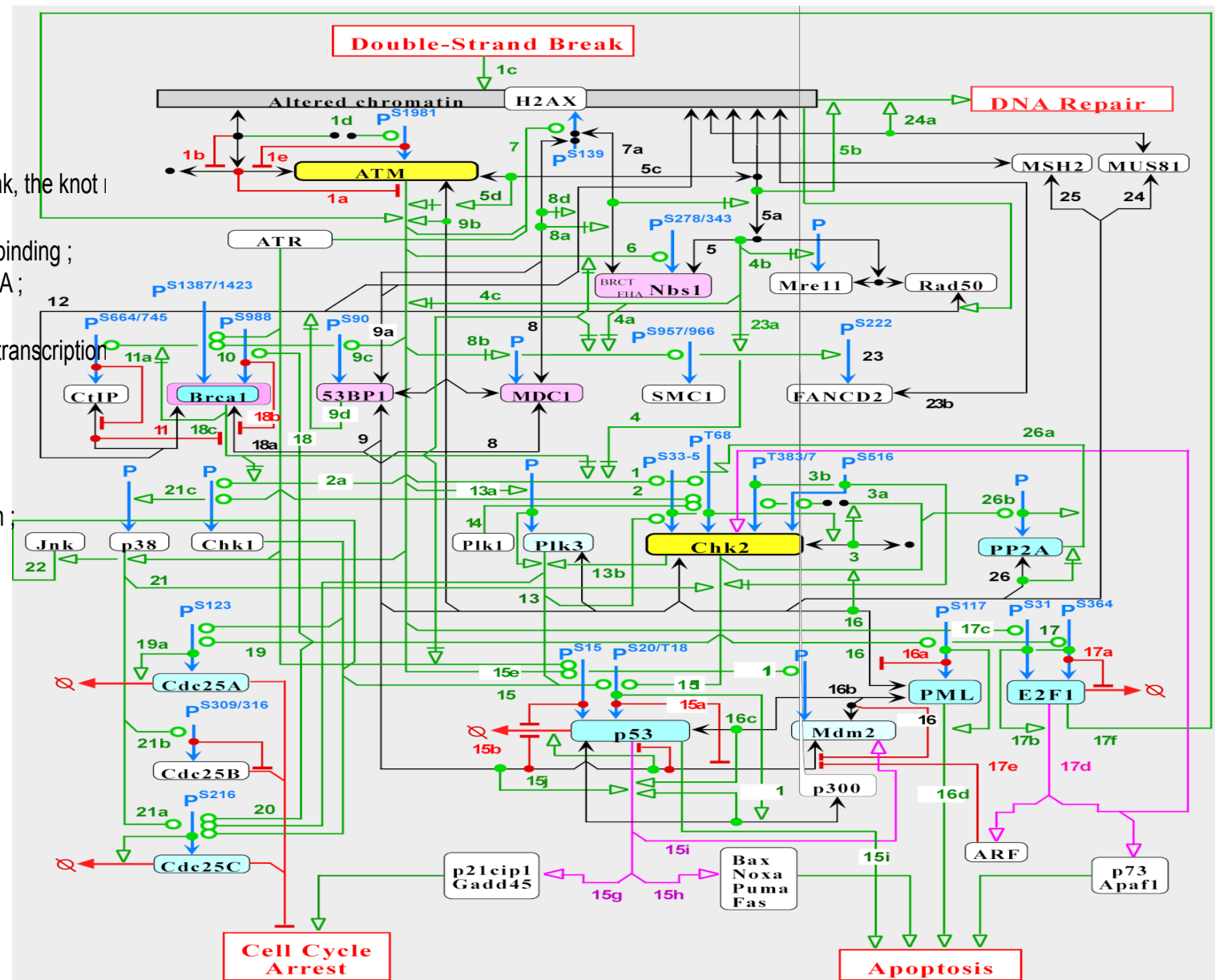


Symbols :

- (a) Proteins A and B could link, the knot is A:B ;
- (b) Multimolecular Complex binding ;
- (c) Covalent Modification of A ;
- (d) Degradation of A ;
- (e) Enzymatic stimulation in transcription ;
- (f) Autophosphorylation
- (g) General Stimulation ;
- (h) Necessity ;
- (i) Inhibition ;
- (j) Activation of transcription ;
- (k) Inhibition of transcription

Translation in default logic:

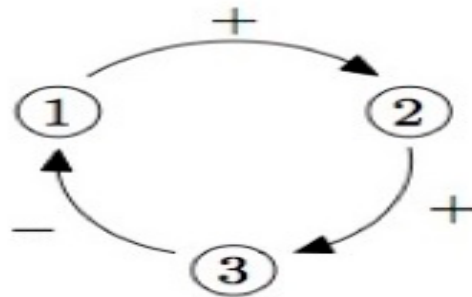
DNA double-strand break-based nonmonotonic logic Doncescu A., and ps 2015. *Emerging Trends in Computational Biology, Bioinformatics, and Systems Biology*. Elsevier.



Problem: Some default theories have no extension.

$$\Delta = (D, W); \quad W = \emptyset; \quad D = \left\{ \frac{\vdots 1}{\vdots 1} \right\}$$

$$\Delta = (D, W); \quad W = \emptyset; \quad D = \left\{ \frac{\vdots 1}{2}, \frac{\vdots 2}{3}, \frac{\vdots 3}{\neg 1} \right\}$$



\Rightarrow Hypothesis logic

3 - Modal logics

- Propositional Calculus (PC) , First Order Logic (FOL) logic gives a truth value in $\{false, true\}, \{0, 1\}$ to any formula: *It's sunny, There is is not sun.*
- Modal logics nuance the values: *I **know** it's sunny ; It's **possible** that it's sunny ; I **think** .. ; **Tomorrow** it will be sunny ; **Elsewhere** it's sunny ; It's **good** that it's sunny*
- We add a *modal operator*, noted **L** (or \Box) to the classical logic.
- We associate to **L** a dual **M** (or \Diamond) defined by **M f** = $\neg L \neg f$.
- The meaning given to couples $\{L, M\}$ depends on the context: (proven, possible), (certain, possible), (know, believe), (obligatory, permitted), (always, one day) ...
- In general case, modalities can be composed: $\neg LL(p \rightarrow (M(p \vee ML \neg q)))$

3 -Modal Logics: Rule and Axioms

Modal logics are defined by *inference rules* and *axiom schemes*

- The basis is the set of rules and axioms of propositional calculus.
- **normal modal** logics include: -
 - *Necessitation rule*: **(N)**: $\frac{\vdash f}{\vdash Lf}$ If f is a « tautology » then Lf is a « tautology ».
 - *Distribution axiom*: **(K)** $\vdash (L(f \rightarrow g) \wedge Lf) \rightarrow Lg$ (An analog of modus ponens)
 $\Leftrightarrow L(f \wedge g) \rightarrow Lf \wedge Lg$
(N) + (K) is the K system (very weak).
- We can also add *reflexivity axiom* $T \vdash Lf \rightarrow f$
(N) + (K) + (T) is the T system.

!!! And that's all in the context of this study.

- We do not add **(4)**: $\vdash Lf \rightarrow LLf$ nor **(5)**: $K \vdash Mf \rightarrow LMf$ because otherwise we lose the notion of dynamic.
- So it is not necessary to compose the modalities (important for the algorithms)

3 - Hypothesis logic \mathcal{H} (ps, Schwind 1988 ... 1993)

\mathcal{H} is a bi-modal logic with 2 modal operators L and $[H]$; $H = \neg[H]\neg$ is the dual of $[H]$

Lf : « f is **proven** » - *The cell **produces** the protein f*
 Hf : « f is **a hypothesis** » - *The cell **give the permission** for attempting to produce f*

- L has the properties of the modal system $T = (N)+(K)+(T)$
- $[H]$ has the properties of the modal system $K = (N) + (K)$

$[H]$ has no reflexivity axiom $[H]f \rightarrow f$ and, there is so far no connections between L and H .

We make this connection by adding the *link axiom schema*:

$$(LI) : \vdash \neg(Lf \wedge H\neg f) \Leftrightarrow Lf \rightarrow \neg H\neg f$$

« *It is impossible to prove f and to assume the hypothesis $\neg f$ at the same time* »

« *It is impossible to produce f and to and to give the permission to destroy f at the same time* »

3 – Hypothesis theories, extensions.

A **hypothesis theory** is a pair $T = \{HY, F\}$ such that F is a set of formula of \mathcal{H} and HY is a set of hypotheses.

- An **extension** E of T is a set $E = Th(F \cup HY')$ such that HY' is a maximal subset of HY consistent with F
« We add to F one of the largest sets of hypotheses to F while remaining consistent »

- An extension E is a **stable extension** if it satisfies the **coherence property**:

$$\forall Hf, \neg Hf \in E \Rightarrow L\neg f \in E$$

Thanks to the link axiom schema, we get:

$$\forall Hf, \neg Hf \in E \Leftrightarrow L\neg f \in E$$

- An extension E is a **ghost extension** if it is not a stable:

$$\exists Hf, \neg Hf \in E \text{ and } L\neg f \notin E$$

- A **sub-extension** a set $E = Th(F \cup HY')$ such that HY' is a non maximal subset of HY consistent with F

- 1) If F is consistent then $T = \{HY, F\}$ has at least one extension.
- 2) Any default theory Δ can be translated in into a hypothesis theory T such that:
 - Any extension of Δ corresponds to a stable extension of T
 - Any stable extension T corresponds to a stable extension of Δ
- 3) Algorithms are **constructive**, not deterministic.

4 - Representation of genetic networks into \mathcal{H}

- \mathcal{H} enables us to use 3 kinds of information: i , Li and Hi .
- Hence, by combining modalities with *negations*, we **can use 5 informations**: $\{i, Hi, H\neg i, Li, L\neg i\}$ and their negations $\{\neg i, \neg Hi, \neg H\neg i, \neg Li, \neg L\neg i\}$ (! $H\neg i \neq \neg Hi$ and $L\neg i \neq \neg Li$)

For genetic networks

- i means that the protein i is **present** in the cell, $\neg i$ means that i is **absent**.
- Li means that i is **produced/activated** by the cell $\neg Li$ means that i is **not produced/activated**
- $L\neg i$ means that i is **destroyed/inhibited** by the cell and $\neg L\neg i$ means that i is **not destroyed/inhibited**.
- Hi (resp. $\neg Hi$) means that the cell **gives (resp. does not give) the permission for attempting to produce i** . Or the cell has (resp. has not) the ability to produce/activate i .
- $H\neg i$ (resp. $\neg H\neg i$) means that **the cell gives (resp. does not give) the permission for attempting to destroy i** . Or the cell has (resp. has not) the ability to destroy/inhibit i .

And an extension gives as many permissions as possible
A permission is effective or not depending on the **environment in the cell** and on \mathcal{H}

4 - Some logic « genetic » properties

- $L\neg i \rightarrow \neg i$ If the network activate/produce i then i is present in the cell (Axiom T)
- $L\neg i \rightarrow \neg i$ If the network inhibit/destroy i then i is absent. (Axiom T)
- $\neg(Li \wedge L\neg i)$ It is impossible to activate and inhibit i at the same time. (From axiom T)
- $\neg(Li \wedge H\neg i)$ Impossible to activate i and to give the authorization to inhibit i at the same time (Link axiom)
- $\neg(L\neg i \wedge Hi)$ Impossible to inhibit i and to give the authorization to destroy i at the same time. (Link axiom)

- All these formulas can be identified to binary propositional clauses.

$$Li \rightarrow i \Leftrightarrow \neg Li \vee i; \quad \neg(Li \wedge H\neg i) \Leftrightarrow \neg Li \vee \neg H\neg i; \quad \dots\dots$$

- For the algorithms, we can consider that the representation is in SAT

5 – Representing general Boolean Dynamical Systems into \mathcal{H}

In NMR 2018 we studied in detail a translation of both positive and negative circuits into H . We extend this translation to any asynchronous BDS.

- Let a BDS characterised by a global transition function $f: X \rightarrow X$: such that:
 $x = (x_1, \dots, x_n) \rightarrow f(x) = (f_1(x), \dots, f_n(x))$ where each function $f_i: X \rightarrow \{0,1\}$ is a local transition function.
- We consider that each x_i is a propositional variable i and therefore that each f_i is a Boolean formula.
- Each f_i is translated into \mathcal{H} : $TR(f_i) = \{ H f_i(x) \rightarrow Li, \quad H f_i(\neg x) \rightarrow L \neg i \}$
- f is translated into : $TR(f) = \cup TR(f_i)$

Example positive 3-circuit : $f(x_1, x_2, x_3) = (\neg x_2, \neg x_3, x_1)$ is translated by :

$$TR(f_1) = \{H2 \rightarrow L\neg 1, \quad H\neg 2 \rightarrow L 1\}$$

$$TR(f_2) = \{H3 \rightarrow L\neg 2, \quad H\neg 3 \rightarrow L 2\}$$

$$TR(f_3) = \{H1 \rightarrow L 3, \quad H\neg 1 \rightarrow \neg L 3\}$$

$$TR(f) = \{H2 \rightarrow L\neg 1, \quad H\neg 2 \rightarrow L 1, \quad H3 \rightarrow L\neg 2, \quad H\neg 3 \rightarrow L 2, \quad H1 \rightarrow L 3, \quad H\neg 1 \rightarrow \neg L 3\}$$

- two stable extensions $E1 = Th(TR(f) \cup \{H1, H\neg 2, H3\})$ and $E2 = Th(TR(f) \cup \{H\neg 1, H2, H\neg 3\})$

- When developing these extensions, we see that they are equivalent to their simplified forms:

$$E1 = \{H1, H2, H\neg 3, L 1, L\neg 2, L 3, \neg H1, \neg H\neg 2, \neg H3, \neg L\neg 1, \neg L 2, \neg L\neg 3\}$$

$$E2 = \{H1, H\neg 2, H3, L\neg 1, L 2, L\neg 3, \neg H\neg 1, \neg H2, \neg H\neg 3, \neg L 1, \neg L\neg 2, \neg L 3\}$$

- We can check that $E1$ and $E2$ are stable extensions because

$$\text{for all } i, \neg Hi \in E1 \Rightarrow L\neg i \in E1 \text{ and } Hi \in E2 \Rightarrow L\neg i \in E2$$

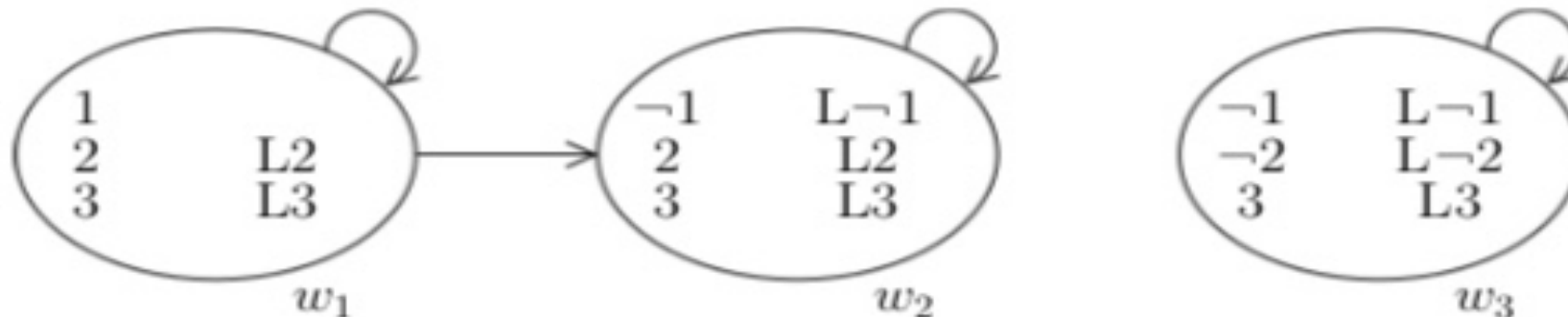
- This is shown by attempting to add to $TR(f)$ each subset of hypothesis and keeping only those among them that are the maximals ones consistent with $TR(f)$
- This can be done using a SAT algorithms

Modal logic: *Kripke semantic* (1963) for normal logics

Kripke structure: $K = (W, R)$

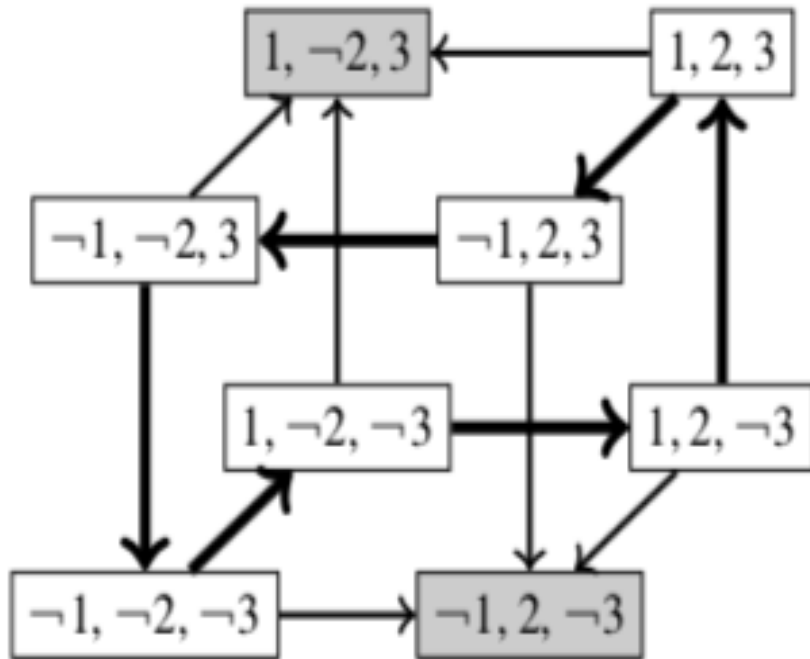
- $W = \{w_k\}$ **universe**, set of **worlds**.
- $R \subseteq W \times W$, **accessibility relation**.
- If $(w_i, w_j) \in R$, w_j is **accessible** from w_i .
- A **Kripke model** is obtained by assigning in every world a truth value to every proposition i .
 - $L f$ is **true in a world w_k** if and only if f is true in all accessible worlds from w_k
 - $M f$ is **true in a world w_k** if and only if f is true in at least one accessible worlds from w_k

Every world is then associated with an interpretation of the propositional logic and thus implicitly with a configuration of a ATG.

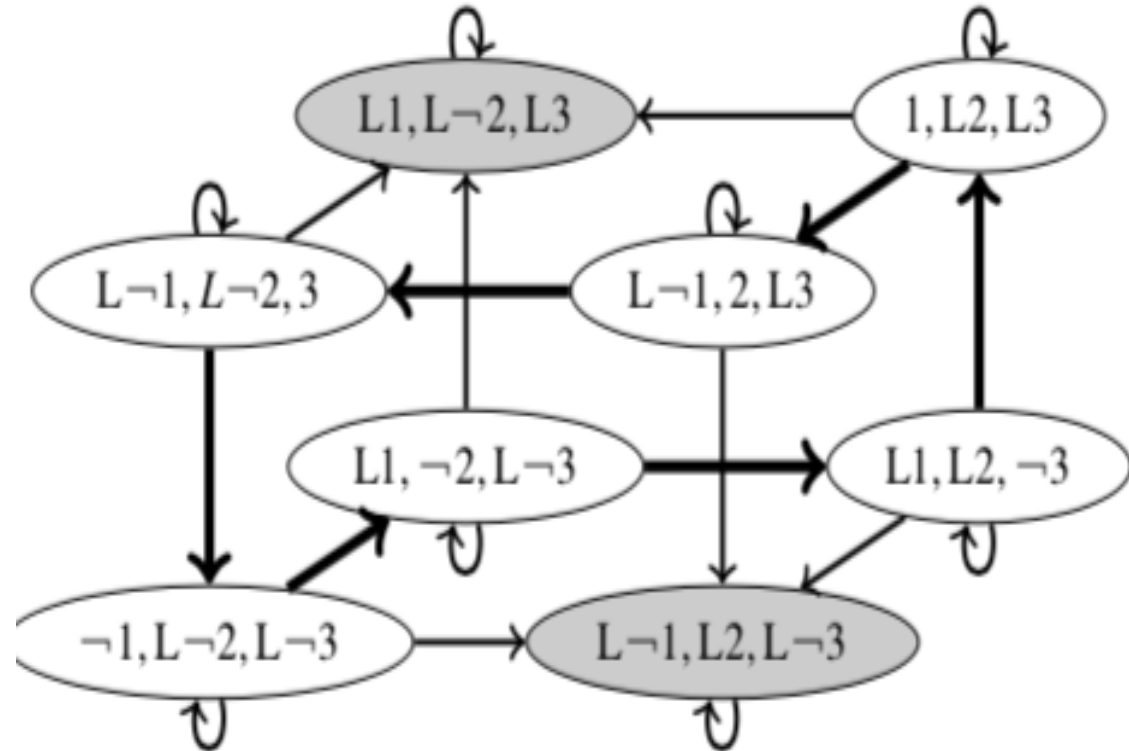


Kripke model of translation of : $f(x_1, x_2, x_3) = (\neg x_2, \neg x_3, x_1)$

- Any configuration of the ATG is associated with a world $w \in W$
- If $(w, w') \in R$ then w and w' are different at most by an i .
- With these conditions translations are Kripke models.

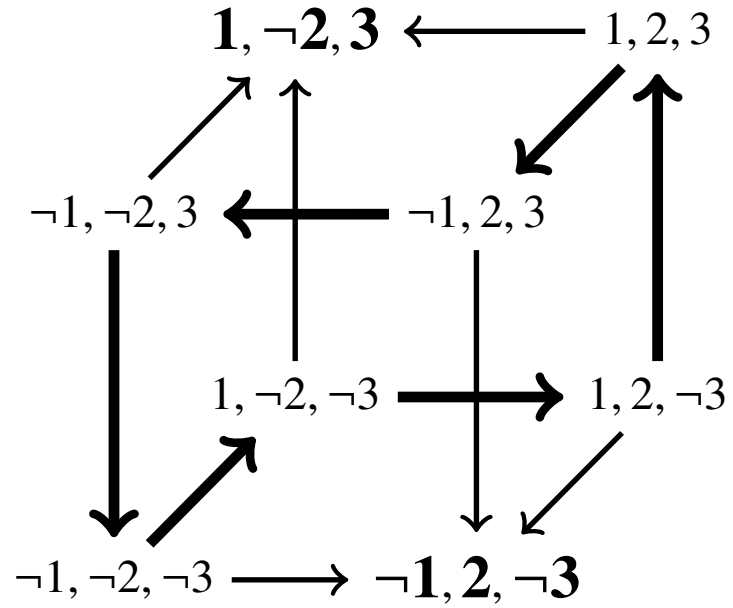


ATG of f

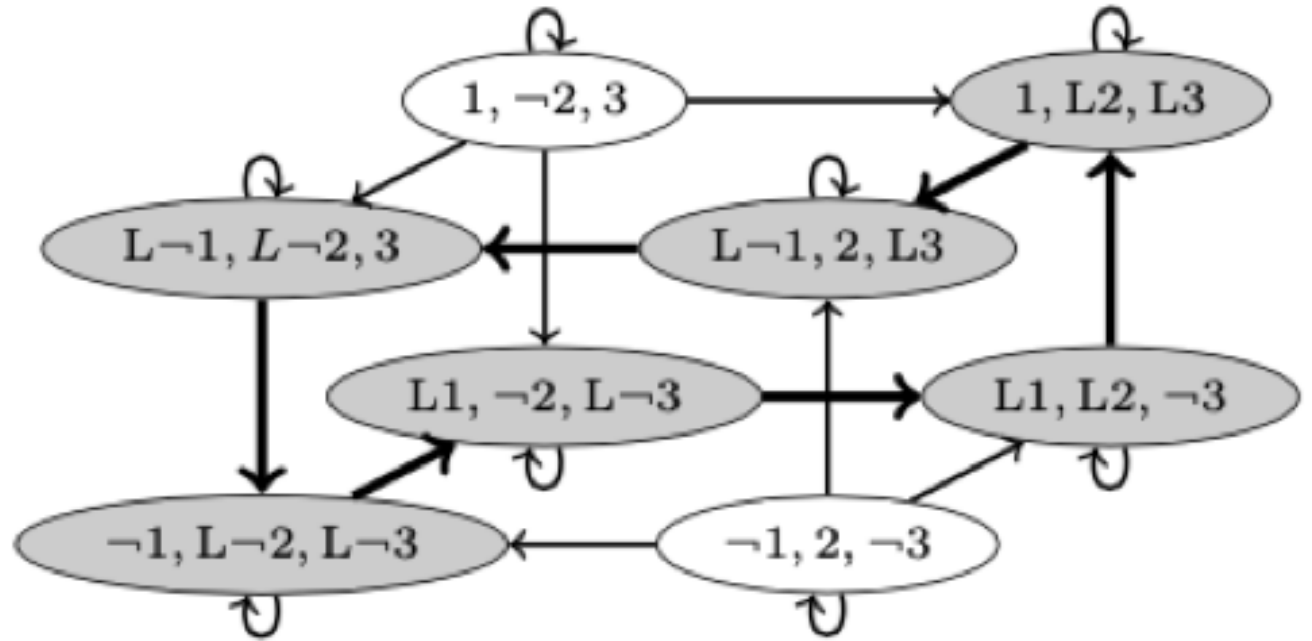


Kripke model of translation of f

Krypke model of translation $g(x_1, x_2, x_3) = (\neg x_3, x_1, x_2)$



ATG of $g(x_1, x_2, x_3) = (\neg x_3, x_1, x_2)$



Krypke model of translation of g

Definition 2. Let E be an extension or a sub-extension:

1. E is complete if, for all $i \in V$, $H i \in E$ or $H \neg i \in E$.
2. A propositional variable $i \in V$ is free in E if $L i \notin E$ and $L \neg i \notin E$. It is fixed otherwise.
3. The degree of freedom of E , denoted by $\deg(E)$, is the number of free propositional variables that compose it.

- $TR(g) = \{H1 \rightarrow L2, H2 \rightarrow L3, H3 \rightarrow L\neg1, H\neg1 \rightarrow L\neg2, H\neg2 \rightarrow L\neg3, H\neg3 \rightarrow L1\}$.

- 6 equivalent ghost extensions of degree 1

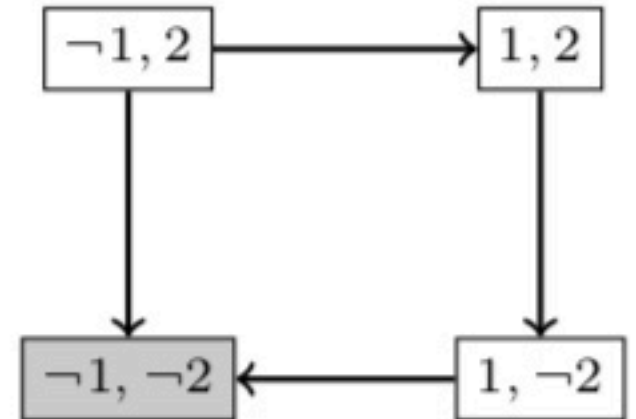
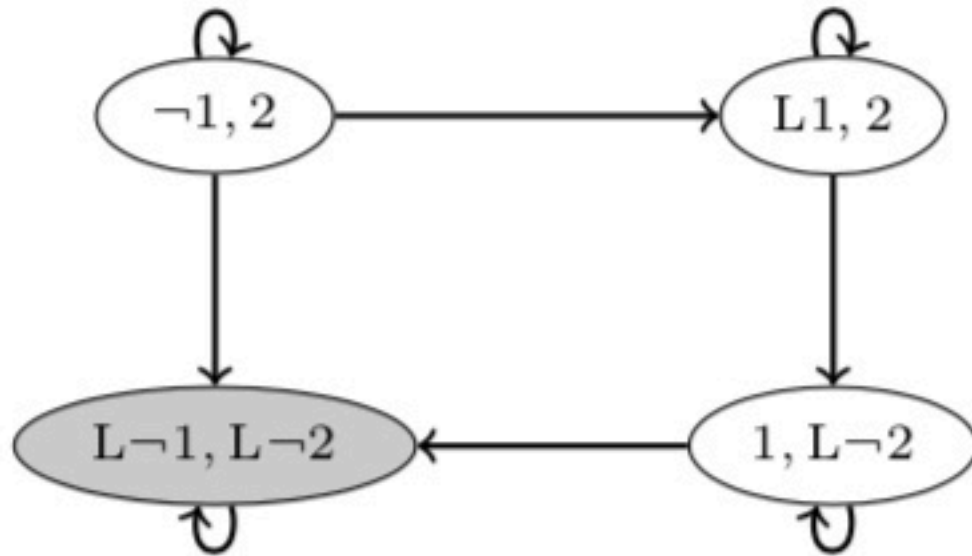
$E1 = \{L 2, L 3\}$, $E2 = \{L \neg1, L 3\}$, $E3 = \{L \neg1, L \neg2\}$, $E4 = \{L \neg2, L \neg3\}$, $E5 = \{L 1, L \neg3\}$, $E6 = \{L 1, L 2\}$.

- $E1, \dots, E6$ are extensions because they are maximal consistent.
- Ghost extensions because in each of them there is a $\neg H i$ (or $\neg H \neg i$) without have $L \neg i$ (or $\neg H i$).
- These extensions represents the stable cycle of \mathbf{g} .

Moreover, there are also two sub-extensions, $E7 = \{1, \neg2, 3\}$ and $E8 = \{\neg1, 2, \neg3\}$ that contain neither $L i$ nor $L \neg i$. Hence all the i is are **free** and their degree of extension is 3.

$$k(x1, x2) = (x2, \neg x1 \wedge x1 \wedge x2)$$

$$TR(k) = \{ H\ 2 \rightarrow L\ 1, \ H\neg 2 \rightarrow L\neg 1, \\ H(\neg 1 \wedge 1 \wedge 2) \rightarrow L\ 2 \), \ H(\neg(\neg 1 \wedge 1 \wedge 2)) \rightarrow L\ \neg 2 \), \\ H(\neg 1 \wedge 1 \wedge 2) \rightarrow L\ 2 \) \ H(1 \vee \neg 1 \vee \neg 2) \rightarrow L\neg 2$$



Properties/theorems

- *A positive circuit is represented by 2 stable extensions.*
- *A negative circuit of size n :*
 - *Has no stable extension, has no extension of degree of freedom 0.*
 - *is represented by a set $2n$ ghost extensions of degree 1.*

The demonstration uses Krypke semantic

Definition 4. Consider a sub-extension, or an extension, E of \mathcal{H} . The projection of E on the system T is the set of formulas of E that do not contain the operator H .

Theorem 3. Let $\mathcal{G}(f)$ be an ATG of function f , and $TR(f)$ be its associated hypothesis theory. The following holds:

1. If $x = \{x_1, \dots, x_n\}$ is a stable configuration of $\mathcal{G}(f)$, then there exists an extension E of degree 0, issued from $TR(f)$, that contains $\{Lx_1, \dots, Lx_n\}$.
2. Let E be an extension of degree 0, issued from $TR(f)$, and w the projection of E . If x is the configuration related to w , then x is stable.

Theorem 4. Let $\mathcal{G}(f)$ be the ATG of function f and $TR(f)$ be its associated hypothesis theory. Every stable cycle C of $\mathcal{G}(f)$ corresponds to a cycle of extensions of degree 1 in $TR(f)$.

6 - Algorithms

- The axiomatic of \mathcal{H} uses only:

- Necessitation rules: $\frac{\vdash f}{\vdash Lf}$ and $\frac{\vdash f}{\vdash [H]f}$
- Distribution axioms for L and [H] equivalents to : $L(f \wedge g) \rightarrow Lf \wedge Lg$ and $[H](f \wedge g) \rightarrow [H]f \wedge [H]g$
- Reflexivity axiom for L: $\vdash Lf \rightarrow f$
- Link axiom : $\vdash \neg(Lf \wedge H\neg f) \Leftrightarrow \neg Lf \rightarrow H\neg f$

So the axiomatic does not have modality composition (particular no LLf and MLf used for axioms (4) and (5)

- For the BDS's and genetic networks, the translation gives only sets of formulas: $\{Hf \rightarrow Li, H\neg f \rightarrow H\neg i\}$ and there is no modality composition.

\Rightarrow We use a fragment of \mathcal{H} such that, all modal elementary formulas have the form Hf or Lf (f is a propositional formula)

\Rightarrow The set of modal formulas is then finite, which allows to switch to SAT.

Technically, if Lf or Hf is a modal predicate we rename the formula f by a new proposition p_f , $f \equiv p_f$

\Rightarrow Each elementary modal formulas could be considered as proposition... In work

6. Algorithms

- The definition of extensions is based on a preferential nonmonotonic approach.
- The algorithms used to calculate extensions are non-deterministic and constructive. There is no need to bother with "loops", to use Clark completion ...
- Algorithms can use a Davis-Putnam procedure or a SAT solver ...
- A Prolog implementation, with Davis-Putnam, has been done for circuits (2-SAT is enough).
- The general case is in work. It is possible to optimize the algorithms (as we are in SAT we have a lot of tools).

Merci !

Pierre Siegel, Camilla Schwind: Modal logic based theory for non-monotonic reasoning. J. Appl. Non Class Logic 3(1): 73-92 (1993).

Camilla Schwind, Pierre Siegel, *A modal logic for hypothesis theory*, Fundamenta Informaticae, 1994, pp. 89--101.

Doncescu, A. , P. Siegel, *DNA double-strand break-based nonmonotonic logic.* , Emerging Trends in Computational Biology, Bioinformatics, and Systems Biology; Elsevier, 2015 pp. 409--427.

Siegel, P., A.Doncescu, V.Risch and S. Sené, *Towards a Boolean dynamical system representation into a nonmonotonic modal logic*, in: Proceedings of NMR'18, 2018, pp. 53--62.

Pierre Siegel, Andrei Doncescu, Vincent Risch, Sylvain Sené, *Logique Modale des Hypothèses, Systèmes Dynamiques Booléens et Réseaux de gènes.* PFIA-JIAF 2020.